HEWLETT
PACKARD

HP 61010A Digital I/O
PC Instruments
Owner's Guide

Safety Summary

The following safety precautions must be observed during all phases of operation of this instrument. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of the instrument. Hewlett-Packard assumes no liability for the customer's failure to comply with these requirements.

Ground the Instrument

To avoid potentially hazardous electrical shock, establish a safety ground before connecting user's circuits. Connect the output cable from the Power Pack to the Digital I/O, and then connect the line cord from the Power Pack to the ac line. Detailed instructions are in the HP PC Instruments System Owner's Manual.

Do Not Exceed Voltage Ratings

Excessive input voltage and current will damage this instrument. Do not exceed ± 10 V maximum per input bit.

Prevent Shorted Connections

To prevent shorted connections when assembling your input and output connectors, do not strip wires back more than 0.15 in. (4 mm). Only use AWG 20 (0.5 mm^2) wire gauge. Only use the form and type of connector originally supplied with the equipment.

Safety Symbols

⚠️   Instruction manual symbol: the product will be marked with this symbol when it is necessary for you to refer to the manual (see What's in this Manual)

⏚   Indicates measuring earth (ground) terminal.

WARNING   The WARNING sign calls attention to a procedure, practice, or the like, which, if not correctly performed or adhered to could result in personal injury. Do not proceed beyond a WARNING sign until the indicated conditions are fully understood and met.

CAUTION   The CAUTION sign calls attention to an operating procedure, or the like, which, if not correctly performed or adhered to could result in damage to or destruction of part or all of the product. Do not proceed beyond a CAUTION sign until the indicated conditions are fully understood and met.

What's in this Manual?

This manual is a supplement to the HP PC Instruments System Owner's Manual. It contains specific information about your HP PC Instruments Digital I/O, Model 61010A. You must read the System Owner's Manual before you read this manual. Warranty and service information is included in the Support Guide in front of your System Owner's Manual.

The System Owner's Manual discusses information that is common to all HP PC Instruments. It also contains specific information about your computer and HP PC Instruments System. If you are a first time user, refer to Table 3 in your System Owner's Manual for the proper reading sequence of your computer and PC Instruments manuals. If you are an experienced user and already have an operating system, you need only read this manual to learn how to operate and program your Digital I/O.

Please insert this manual in the same hardcover binder as your System Owner's Manual. Here is a brief description of the contents of each chapter in this manual:

Chapter 1 · Product Description
Briefly describes the Digital I/O, gives its specifications, and lists the items that you receive with it.

Chapter 2 · Trying Out Your Instrument
Gives simple step-by-step instructions that let you quickly perform some operations with nothing connected to the front panel.

Chapter 3 · Manual Instrument Control
Gives detailed operating information not covered in the simplified instructions of Chapter 2.

Chapter 4 · Front Panel Connections

⚠ Explains how to connect the Digital I/O to your
application.

Chapter 5 · Programming With BASIC

Describes how to control the Digital I/O with a
program. All programming statements for the Digital
I/O are explained. Simple programming examples are
also included.

Appendix A · Programming Statement Summary

Lists all programming statements that apply to the
Digital I/O.

Appendix B · Verification Procedures

Describes verification procedures that you can use to
verify the proper operation of the Digital I/O if you
suspect an instrument malfunction.

Appendix C · Error Messages

Lists all error messages that apply to the Digital I/O.

Appendix D · HP 14802A Terminal Block Connections

Describes the optional Terminal Block and identifies all
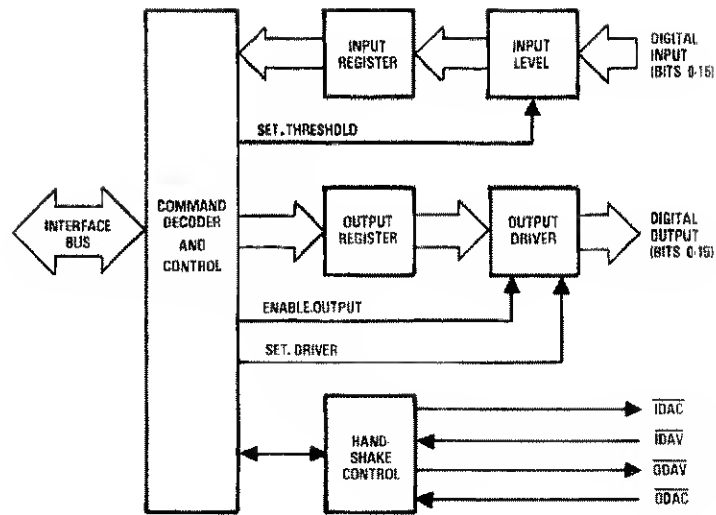screw terminal assignments.

1

Product Description

Introducing the HP PC Instruments Digital I/O

The HP PC Instruments Digital I/O is an instrument that can send and receive 16-bit data words through the input and output connectors on the front of the unit. It does this under the control of a computer that is equipped with PC Instruments software and an interface card. The System Owner's Manual tells you all you need to know about the required software and interface card for your computer. Chapters 2 and 3 of this manual explain how to operate the Digital I/O "manually" using the Soft Front Panel (computer display) while Chapter 5 describes programming statements you can use to control it from BASIC.

Figure 1-1 is a block diagram that illustrates the two primary functions to the Digital I/O. These functions are: Digital Input and Digital Output. The 16 data bits of both the Digital Input and Digital Output can be addressed individually or simultaneously as a word. In this way you can specify variable length data words up to 16 bits long as well as their location on the input and output connectors.

The input data bits are compatible with all logic levels. Programmable circuitry on the instrument allows you to specify an input threshold level. The output data bits can be programmed to be compatible with either TTL or open collector logic levels. In addition to the 16 data bits, the input connector and the output connector also contain two control signals and four digital commons each. Complete specifications are given in Table 1-1.

INPUT REGISTER

INPUT LEVEL

DIGITAL INPUT (BITS 0-15)

SET.THRESHOLD

INTERFACE BUS

COMMAND DECODER AND CONTROL

OUTPUT REGISTER

OUTPUT DRIVER

DIGITAL OUTPUT (BITS 0-15)

ENABLE.OUTPUT

SET. DRIVER

HAND-SHAKE CONTROL

$\overline{IDAC}$

$\overline{IDAV}$

$\overline{ODAV}$

$\overline{ODAC}$

Figure 1-1. Digital I/O Block Diagram

Two basic operating modes are available for I/O data transfers: synchronous and asynchronous. Synchronous operation requires that some other instrument or hardware source must communicate with the Digital I/O. This two-way communication is referred to as "handshaking". When sending data to an external device, the Digital I/O signals the external device that data is available. The external device completes the handshake by returning a signal that indicates it has received the information. When receiving data from an external device, handshaking requires that the external device must first signal the Digital I/O that data is valid. To complete the handshake, the Digital I/O will return a signal to indicate it has successfully received the data.

In asynchronous mode, the computer alone controls the transmission of data on the Digital I/O's input and output bits. Any data that is sent by the computer immediately appears on the output bits. Data that appears on the input bits is immediately read into the computer. Data transmission is not dependent on any control signals.

Items Supplied

In addition to this manual, check that you have received the following items with your Digital I/O:

Power Pack - an ac power transformer with an attached one metre cable. The transformer type was determined by country of destination. Chapter 2 of the System Owner's Manual lists the different types and their part numbers.

Power Cord - connects the Power Pack to an ac source. Plug type was determined by the country of destination. Chapter 2 of the System Owner's Manual lists types and part numbers.

Instrument Interconnect Cable (8120-4631) - connects your instruments to the system interface. Refer to Chapter 2 of the System Owner's Manual.

I/O Terminal Connectors (1252-0928) - two connectors (one male, one female) with covers are for application connections to the Digital I/O. The third male connector is used for verification.

Update Pages - if applicable, update pages are included. Replace the obsolete pages with the new ones before you use this manual.

Optional Hardware

Accessory Block (HP 14802A) - allows easy screw termination connections to the Digital I/O (see Appendix D).

Table 1-1. Specifications

User Connections:

16 Input data bits
2 Input data control signals
16 Output data bits
2 Output data control signals

Input Data Characteristics:

Input voltage range = ± 10 V max.
Input impedance = 100 k with internal pull-up resistor to +5 V

Logic threshold (programmable to ± 10 V)
Resolution = 80 mV
Accuracy = ± 160 mV

Output Data Characteristics:

TTL mode
V_{ol} = 0.4 V max. @ I_{ol} = 16 mA max.
V_{oh} = 2.4 V min. @ I_{oh} = −4 mA max.

Open collector mode
V_{ol} = 0.4 V max. @ I_{ol} =16 mA max.
V_{ol} = 0.7 V max @ I_{ol} = 40 mA max.
V_{oh} = 12 V max. (with customer supplied external pull-up resistor to external supply)

Table 1-1. Specifications (continued)

Control Signal Characteristics:

$\overline{\text{ODAV}}$; $\overline{\text{IDAC}}$
Same as output data bit

$\overline{\text{ODAC}}$; $\overline{\text{IDAV}}$
$V_{il} = 0.0$ V to 0.4 V
$V_{ih} = 2.4$ V to 5.0 V
(with internal 10 k Ω pull-up resistor to +5 V)
Minimum pulse width $= 10\mu s$
Rise and fall time $< 1\mu s$ (10% to 90%)

Data Transfer Time:

< 50 ms (using an OUTPUT or MEASURE statement)

Operating Temperature Range:

0°C to 40°C

Storage Temperature Range:

−40°C to +80°C

Dimensions:

Length = 295 mm (11.62 in.)
Width = 212 mm (8.35 in.)
Height = 64.5 mm (2.54 in.)

Weight:

1.76 kg. (2.78 lbs.)

Table 1-1. Specifications (continued)

Factory Defaults:
  Digital input
    Label =DIG.IN.01 (DIG.IN.02 for 2nd instrument)
    Base = decimal
    Complement = 2's
    Logic sense = positive
    Length of bits = 16
    Start bit = 0
    Threshold = 1.36 V
    Handshake = disabled
    Update = auto

  Digital output
    Label = DIG.OUT.01 (DIG.OUT.02 for
     2nd instrument)
    Output value = 0
    Base = decimal
    Logic sense = positive
    Length of bits = 16
    Start bit = 0
    Driver = TTL
    Handshake = disabled
    Output = disabled

2

Trying Out Your Instrument

Introduction

The following step-by-step procedure allows you to perform some simple instrument operations. This procedure is especially suitable for first time users who want to quickly become familiar with the basic operation of the Digital I/O. Chapter 3 contains additional operating information that you can use once you have learned the basics in this chapter.

The Procedure

Trying out your instrument consists of pointing to and selecting various interactive fields on the Soft Front Panel. Before you try out your Digital I/O, you should have: connected it to the interface, applied power, loaded the operating system, renamed (or erased) HPSTATE.HPC to return the instrument to its factory default settings, and run PANELS as explained in Chapters 2 and 3 of your System Owner's Manual.

NOTE

If the output of your instrument is connected to an application, be aware that following these instructions will set output bits 2, 5, and 6 high. You may want to disconnect your output connector.

Digital Output   **Step 1** - If DIG.OUT.01 is not already in the Interactive
Instrument Window point to and select DIG.OUT.01 from
the labels listed in the System View Window (see Figure
2-1). If you have many instruments connected to the
interface, use the ▓ROLL UP SYS VIEW▓ softkey to view them
all. When you select a label from the list, the ACTIVE
indicator on the front of the Digital I/O that is associated
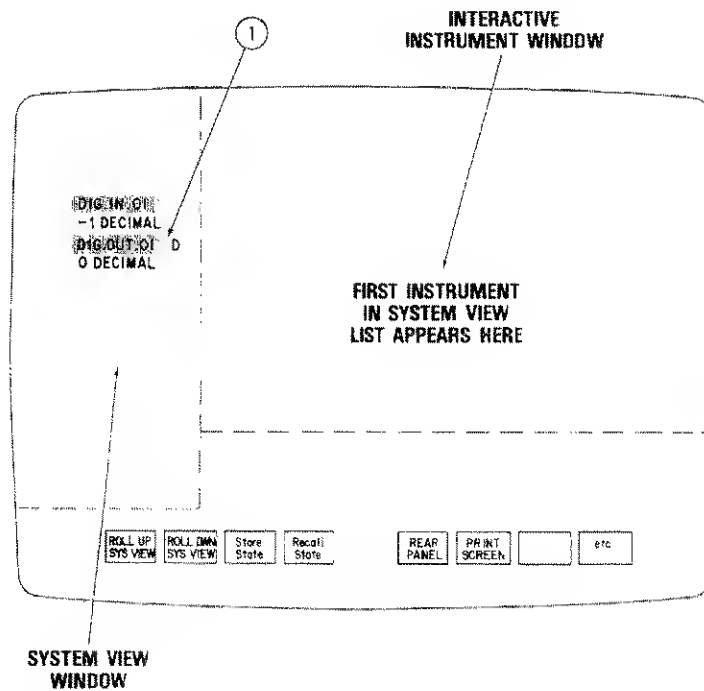with that label lights up.

Figure 2-1. Select Digital Output

As shown in Figure 2-1, your Digital I/O places two labels in the System View Window list; one for the Digital Input and one for the Digital Output. DIG.OUT.01 is the factory default label for the Digital Output. The letter D after the label indicates that the output is disabled at this time. The present output value is displayed in the line directly below the label in the System View Window. This value can be displayed in decimal (DEC), octal (OCT), or hexadecimal (HEX) format. Binary values are displayed in octal format. If you have more than one Digital I/O connected to your interface, each additional Digital Output will be assigned a sequentially numbered default label (i.e. DIG.OUT.02, DIG.OUT.03, etc).

Step 2 · Once you select DIG.OUT.01, the DIG.OUT.01
information in the System View Window disappears, and
a front panel representation of the Digital Output appears
in the Interactive Instrument Window (see Figure 2-2).
Notice that 0 is the current output value. To change this
and output a different value, first point to and select the
output field.
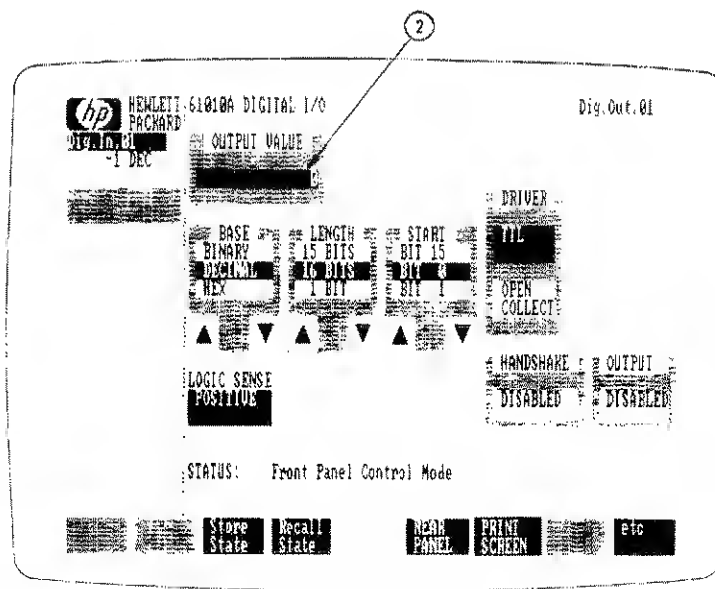

Figure 2-2. Select the Output Field

Step 3 · When the output field is selected, the Keyboard Entry Menu softkeys appear on the screen (see Figure 2-3). Use the keyboard keys to position the cursor and type in the value 100. If you make a mistake, press the Cancel Command softkey to return to the original value and try again. After you have typed in the new value, press [Return] .
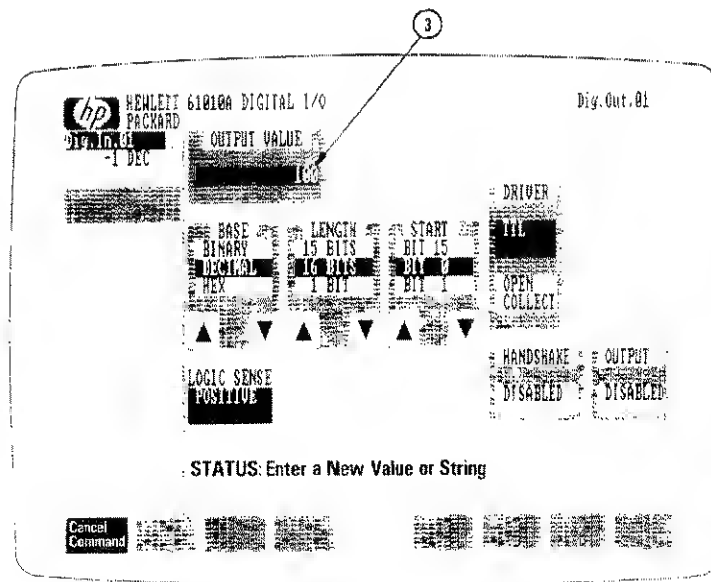

Figure 2-3. Enter a New Output Value

Step 4 - After you press ⌷Return⌷ , the original softkeys return to the screen (see Figure 2-4). Because the output field indicates DISABLED, your new entry did not set any output bits on your instrument yet. The output must be enabled before any output bits can be set. Point to and select DISABLED to enable the output. (The output field toggles between ENABLED AND DISABLED.)
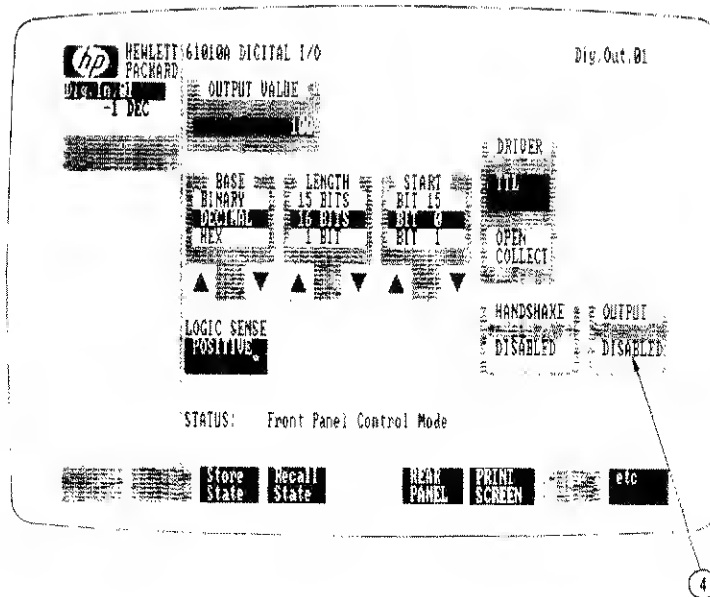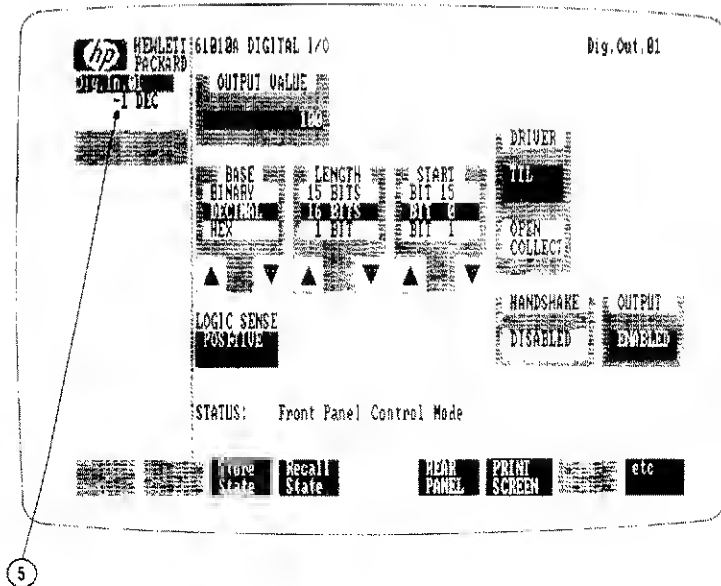

Figure 2-4. Enable the Output

The Result - With the output ENABLED, the output bits that correspond to a decimal value of 100 are set (see Figure 2-5). In this case, output bits 2,5, and 6 are set high.
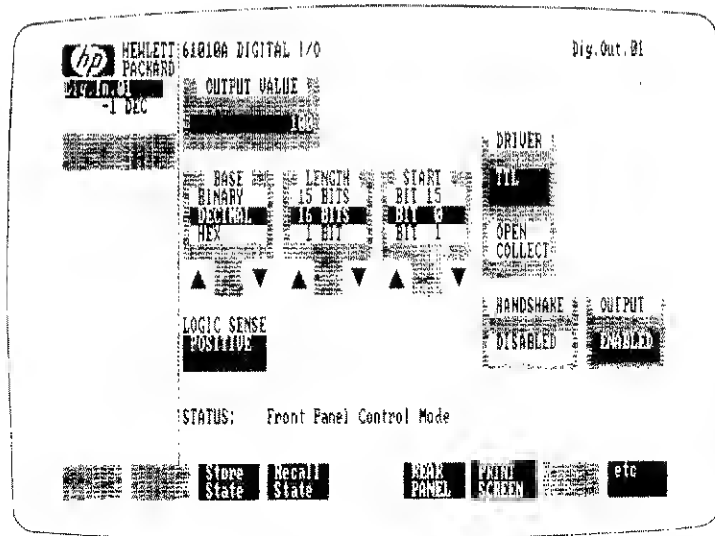
Figure 2-5. Output Bits 2,5, and 6 Set High

Digital Input

Step 5 - To try out DIG.IN.01, point to and select DIG.IN.01 from the labels in the System View Window (see Figure 2-6).

Figure 2-6. Select Digital Input

As shown in Figure 2-6, DIG.IN.01 is the factory default label for the Digital Input. The present input value is displayed on the line directly below the label in the System View Window. This value can be displayed in decimal (DEC), octal (OCT), or hexadecimal (HEX) format. Binary values are displayed in octal format. If you have more than one Digital I/O connected to your interface, each additional Digital Input will be assigned a sequentially numbered default label (i.e. DIG.IN.02, DIG.IN.03, etc).

Step 6 · When you select DIG.IN.01, the DIG.IN.01 information in the System View window disappears, and a front panel representation of the Digital Input replaces the Digital Output in the Interactive Instrument Window. The default base for DIG.IN.01 is decimal. Notice that with the input connector disconnected, a −1 appears on the display (see Figure 2-7). This is because the Digital Input is reading all 16 input bits as being high due to internal pull-up resistors and the fact that the logic sense is positive. The −1 is the 2's-complement representation of all 16 input bits being high. To illustrate how the Digital Input can format the input data on the display, point to and select 2's in the set compl field. (The set compl field toggles between 2's and UNSIGN).
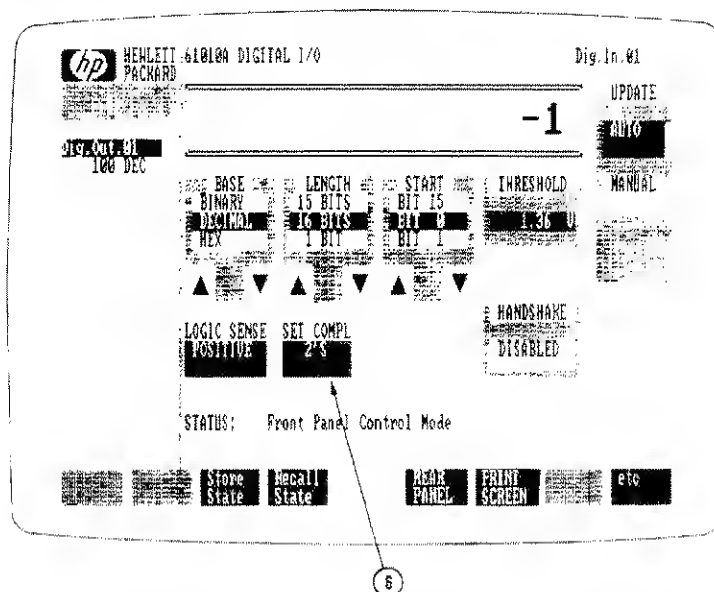
Figure 2-7. Change Complement to Unsigned

Step 7 - The input display changed to 65535, which is the unsigned representation for all 16 input bits being high. Now point to and select the ▼ arrow to change the base to BINARY (see Figure 2-8). Each time you point to and select the ▼ arrow, the top choice in the list rolls down into the center position. The choice at the bottom of the list rolls around until it appears at the top of the list again. (The choice that is currently active stays bright as it rolls around in the list).
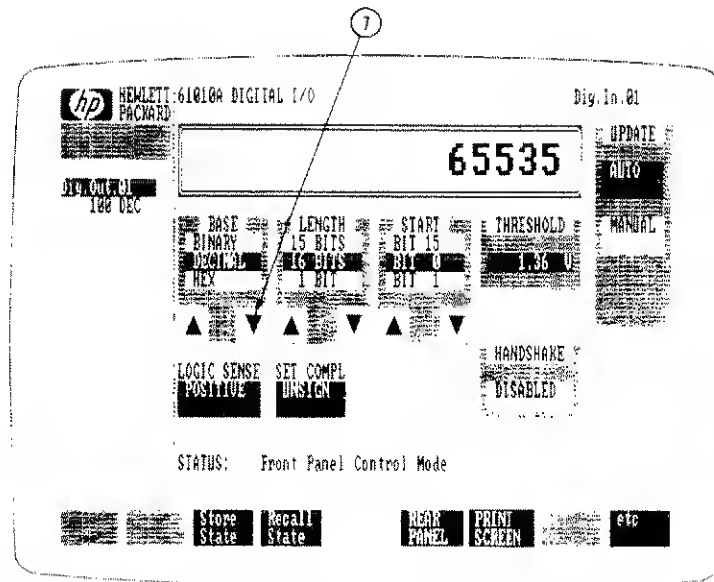
Figure 2-8. Change the Base to Binary

Step 8 · When the binary base appears in the center position on the list, point to and select the center position to activate the binary base (see Figure 2-9).
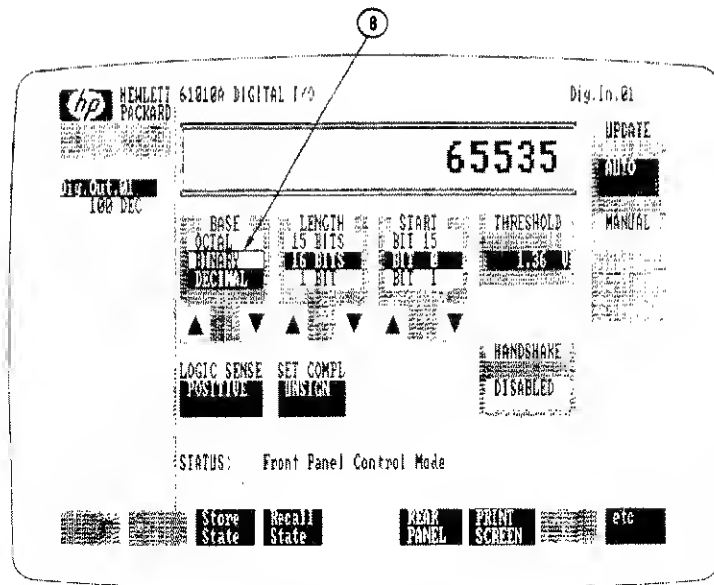
Figure 2-9. Activate the Binary Base

The Result · The input display now indicates
1111111111111111 which shows all 16 bits high (see Figure
2-10). Input bit 0 (the LSB) is on the right, and input bit
15 (the MSB) is on the left. Because binary is the selected
base, the complement field disappeared from the screen
because it no longer applies. To try out a different
instrument at this time, go to Chapter 2 of its manual.
Otherwise, press the etc. and then the EXIT softkey.
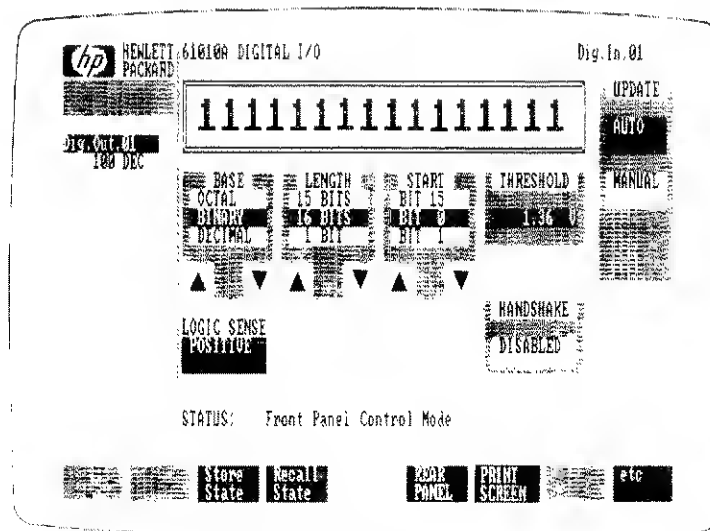
Figure 2-10. All Input Bits High

What to do Next

Now that you have finished trying out your instrument,
the next thing you do depends upon the type of user you
are. If you are an experienced user and are already
familiar with the System Owner's Manual, read the
remaining chapters in this manual. If you are a first time
user of the PC Instruments System, you should follow the
reading sequence given in Table 3 of the System Owner's
Manual.

3

Manual Instrument Control

Introduction

You may have some idea of how to manually operate the Digital I/O from the procedure given in Chapter 2. That chapter gave you a brief overview of some of the Soft Front Panel controls of your instrument. This chapter gives you detailed information about inputting (reading) and outputting (writing) data on your Digital I/O. Separate descriptions are given for each of these two main functions of your instrument, which are referred to as Digital Input and Digital Output. Chapter 4 of the System Owner's Manual describes the Soft Front Panel features (including softkeys) that are common to all instruments.

Selecting the Digital Input

The Digital I/O must be displayed in the Interactive Instrument Window before you can manually operate it. Only the Digital Input or the Digital Output can be operated at one time. If the Digital Input is not already displayed in the Interactive Instrument Window, point to and select its label in the System View window.

Figure 3-1 shows the factory default settings that appear when the first Digital Input in your PC Instruments System is selected. These defaults are: decimal base, 2's-complement, positive logic sense, 16-bit length, 0 start bit, 1.36 V threshold, handshake disabled, and auto update. The default label (DIG.IN.01) appears in the upper right corner of the Interactive Instrument Window. The Interactive Instrument Window can display either the front panel or rear panel of the Digital Input. Use the FRONT PANEL softkey to display the front panel. This is the panel by which you operate the Digital Input. If the front panel that appears on your computer's display does not

look like Figure 3-1, it means that a previous user has
changed the default settings.
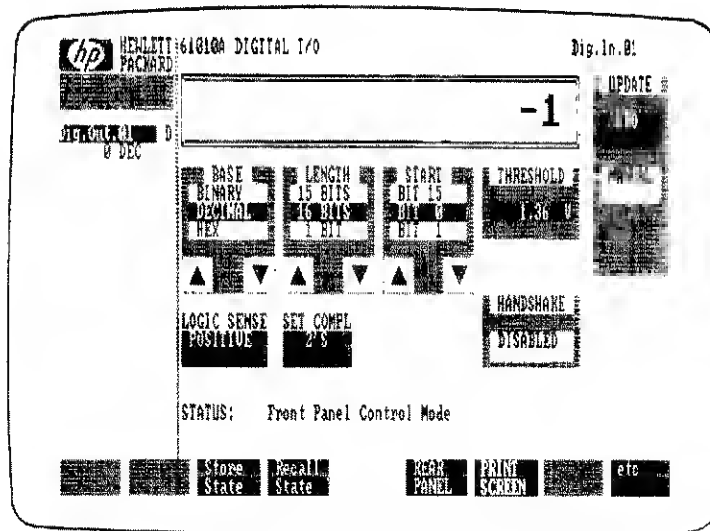
Figure 3-1. Digital Input Default Settings

Operating the Digital Input

The present input value is displayed in the large display
area at the top of the Interactive Instrument Window (see
Figure 3-1). Below this area are the interactive graphic
fields. With these fields you can modify the input data for
convenient interpretation when it is displayed on the top
of your screen. Some of these fields are also used when
operating your Digital Input synchronously as explained
later in this section.

Asynchronous Operation

Operating your Digital Input asynchronously means that
the display area of the Interactive Instrument Window is
updated either automatically or manually to reflect the
current state of the input bits. Data representation is

determined by the settings of the base, complement, length, start, threshold, and logic sense fields. If you change any of these settings (except base), the display area on your screen becomes blank. This indicates that the way the present input data is displayed is invalid. The display area remains blank until the next time input data is read by the Digital Input.

BASE · contains four choices (one is hidden), that allow you to represent input data on the display area of the Interactive Instrument Window in decimal, binary, octal, or hexadecimal format. The current selection is always bright. If the desired base is not bright, use the arrows at the bottom of the field to move the desired base to the center location. When the desired base appears in the center location, point to and select it to activate the base. The newly selected base will now be bright to indicate that it is the current selection.

SET COMPL · when the decimal base is pointed to and selected, the set compl field appears on your screen as shown in Figure 3-1. When you point to and select this field, it toggles between 2's and UNSIGN. The field always indicates the complement that is currently active. Selecting this field will activate the hidden choice. When 2's-complement is active, your input data is displayed as positive and negative decimal values on the display area of the screen. The left-most bit of any specified data field automatically becomes the sign bit. Unsigned format displays your input data as positive decimal values only. If a base other than decimal is selected, the set compl field does not appear on the screen.

To help you understand how the base selection affects the way input data is displayed, lets assume that input bits 0 through 12 are set low, and bits 13 through 15 are set high. Depending on the base, the input data is displayed as follows:

Decimal (2's compl) = −8192
Decimal (unsigned) = 57344

Binary = 1110000000000000
Octal = 160000
Hexadecimal = E000

LOGIC SENSE · formats the incoming data as either positive true logic, or a negative true logic (input bits inverted). When you point to and select this field, it toggles between POSITIVE and NEGATIVE. The field always indicates which logic sense is currently active. The polarity of the input control signals are unaffected by action of this field.

In the previous example the logic sense was assumed to be positive. Lets assume that the input bits are set the same as before, only this time the logic sense is negative. Depending on the base, the input data is displayed as follows:

Decimal (2's compl) = 8191
Decimal (unsigned) = 8191
Binary = 0001111111111111
Octal = 017777
Hexadecimal = 1FFF

LENGTH · contains 16 choices (13 are hidden) that allow you to specify the size of the input data segment (number of input bits) that will be read by the Digital Input. Thus, the length can vary in size from 1 bit to 16 bits. Use the arrows at the bottom of the field to move the desired choice into the center location. When a selection appears in the center location, it automatically becomes active. To prevent errors, always specify the length of the field before you specify the start bit. If errors do occur, a message in the Status Window will identify the type of error that has occurred.

START · contains up to 16 choices (13 are hidden) that let you specify the starting location or LSB of your input data segment. Because this field works in conjunction with the length field, your start choices are limited by the length of your field. If you specify a length of one bit, you have a

choice of 16 start bits. If you specify a length of 16 bits, you only have a choice of one start bit (bit 0). This prevents the previously specified length of the data segment from being truncated.

The following illustration shows you which input bits are read when a length of 4 and a start bit of 8 is specified. This example shows how the length corresponds to the number of bits to the left of and including the start bit. Note that the start bit cannot be a number greater than 12, otherwise the length of the data field would be truncated. Unspecified input bits will not be read.

$$\underset{\text{bit }15}{\text{X X X X}}\ \boxed{\text{X X X X}}\ \underset{}{\text{X X X X X X X X}}\underset{\text{bit }0}{}$$

THRESHOLD - is a voltage level that defines the crossover point or dividing line between two logic states. The default threshold voltage level is 1.36 V. How the Digital Input interprets the logic states depends upon the setting of the logic sense.

When the logic sense is positive, voltages above the threshold level will be interpreted as a logic "high"; voltages below the threshold level will be interpreted as a logic "low". When the logic sense is negative, voltages above the threshold level will be interpreted as a logic "low"; voltages below the threshold level will be interpreted as a logic "high".

To change the threshold, point to and select the threshold voltage field. There are two ways you can change the threshold. You can select the New Value softkey from the keyboard and enter a new value, or you can modify the existing value by using the keyboard softkeys that correspond with the ones shown on your screen. Both methods are explained in Chapter 4 of the System Owner's Manual. Valid entries are voltages in the range of −10.24 V to +10.16 V with a resolution of 0.08 V.

UPDATE - determines the moment when the input data is read by the Digital Input. When AUTO is selected, the input data bits are continuously monitored and the display area in the Interactive Instrument Window is updated accordingly. When MANUAL is selected, a START field immediately appears under the MANUAL field (see Figure 3-2). Data is read into the Digital Input only when the START field is selected.

When the Digital Input appears in the System View Window, the input display will not be updated if MANUAL update is selected. The letter "M" appears next to the Digital Input's label in the System View Window to remind you that it is in manual update mode. Select AUTO update if you want the input display to update when the Digital Input is in the System View Window.

Synchronous Operation

Operating your Digital Input synchronously requires that an external device signals the Digital Input to indicate that input data is valid. After the Digital Input reads the data, it returns a signal to the external device to indicate that it has successfully received the information. Two control signals — Input Data Available and Input Data Accepted — are available at the input connector, both of which must be connected to your external equipment (refer to Chapter 4). Once the connections are made, you are ready to operate your Digital Input synchronously.

The front panel graphics already discussed (i.e. base, set compl, logic sense, length, start, threshold, and update fields) operate exactly the same way in synchronous mode as they do in asynchronous mode. The only additional fields that are used when operating your instrument synchronously are the handshake and reset fields (see Figure 3-2).
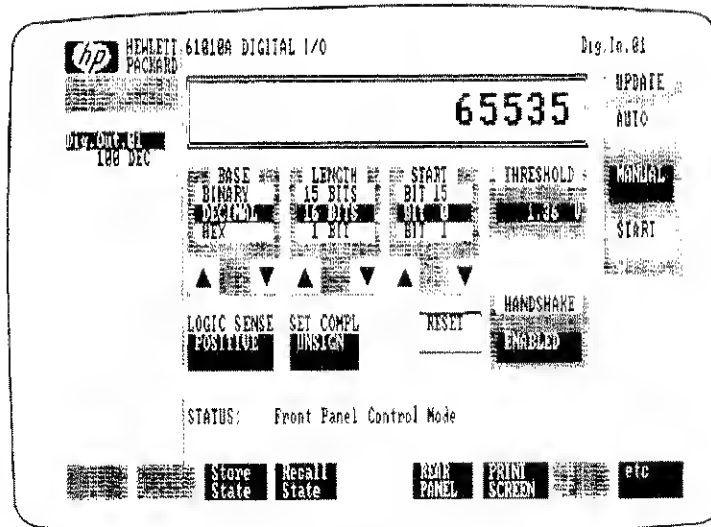
Figure 3-2. Handshake Enabled

HANDSHAKE - must be ENABLED to operate your instrument synchronously. The handshake field toggles between ENABLED and DISABLED. Point to and select DISABLED to enable the control signals that allow handshaking with your external equipment.

RESET - when the handshake is enabled, the reset field appears on the screen to indicate that the Digital Input is waiting for a handshake sequence to begin. The handshake sequence actually begins when the external device sends an Input Data Available (IDAV) signal to indicate that external data is available. When this occurs, the reset field no longer appears on the screen. Point to and select the reset field when it appears on the screen to discontinue the waiting process if an IDAV signal is not received.

When manual update is in effect, the reset field appears on the screen only after you select the start field. When auto update is in effect, the reset field always appears on the screen. This is because even after $\overline{\text{IDAV}}$ is sent or when you select reset to discontinue waiting, the Digital Input will now wait for the *next* $\overline{\text{IDAV}}$ to be sent.

When the Digital Input appears in the System View Window in manual update mode with the handshake enabled, the letter "W" can appear next to the Digital Input's label. This letter appears whenever the Digital Input is waiting for new input data from the external device. After the Digital Input receives the new data, the input display is updated and the letter "M" will appear next to the Digital Input's label.

Figure 3-3 shows the timing relationship between the Input Data Available ($\overline{\text{IDAV}}$) signal that must be provided by the external device, and the Input Data Accepted signal ($\overline{\text{IDAC}}$) from the Digital Input.
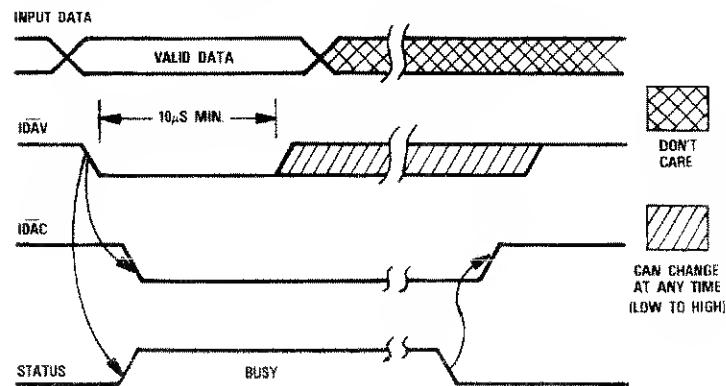
Figure 3-3. Input Handshake Timing Diagram

The high-to-low transition of $\overline{\text{IDAV}}$, (when pulled low by the external device alerts the Digital Input that data is available. The input data must be stable before $\overline{\text{IDAV}}$ is

sent, and $\overline{IDAV}$ must be set low for a minimum of $10\mu s$. The status register on the Digital I/O indicates that the Digital Input is now busy reading data. When the Digital Input sets $\overline{IDAC}$ low, $\overline{IDAV}$ can go high again at any time as long as the minimum pulse width requirement has been satisfied. When the Digital Input has accepted the input data, the status register changes state and $\overline{IDAC}$ is set high again. This signals the external device that the Digital Input is ready for another data transaction.

If a second high-to-low transition of $\overline{IDAV}$ is encountered during the time that $\overline{IDAC}$ is being pulled low, a message on the Soft Front Panel will alert you to the fact that your input data was "overrun", that is, changing faster than the computer can accept it. This means that your input data was changed before it had been completely read into the computer from the Digital Input.

Selecting the Digital Output

If the Digital Output is not already displayed in the Interactive Instrument Window, point to and select its label in the System View window.

Figure 3-4 shows the factory default settings that appear when the first Digital Output in your PC Instruments System is selected. These defaults are: decimal base, 16 bit length, 0 start bit, TTL driver, positive logic sense, handshake and output disabled. The default label (DIG.OUT.01) appears in the upper right corner of the Interactive Instrument Window. The Interactive Instrument Window can display either the front panel or rear panel of the Digital Output. Use the FRONT PANEL softkey to display the front panel. This is the panel that lets you operate your instrument. If the front panel that appears on your computer's display does not look like Figure 3-4, it means that a previous user has changed the default settings.
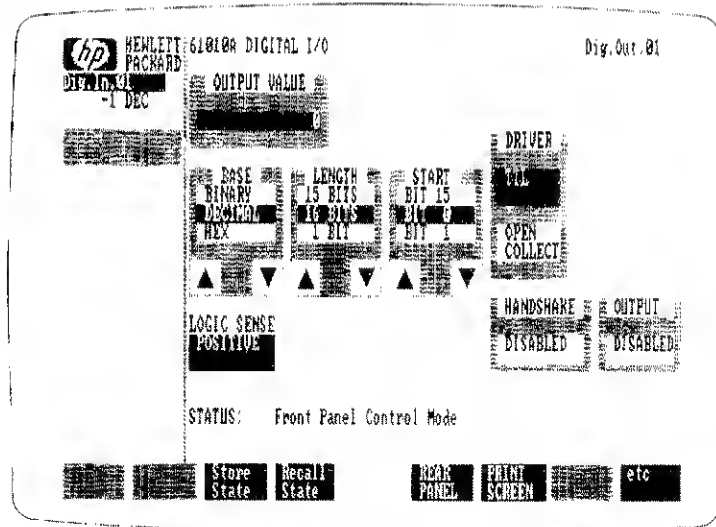
Figure 3-4. Digital Output Default Settings

Operating the Digital Output

The present output value is displayed in the output value field at the top of the Interactive Instrument Window (see Figure 3-4). This field is also used to enter new output values. Below this field are other interactive graphic fields that you can use to simplify data entry into the output field, and make the output bits compatible with your application. Some of these fields are also used when operating your Digital Output synchronously as explained later in this section.

Asynchronous Operation

Operating your Digital Output asynchronously means that when data is entered into the output value field and the output is enabled, it automatically outputs the data. Before entering data into the output value field, you must set the base, length, start, and logic sense fields to the proper settings. If you change any of these settings (except base) *after* you have entered an output value, that value becomes

invalid and the output value field becomes blank. This indicates that you must now enter a new output value.

NOTE

Just because the output value field becomes blank does not mean that the output data has changed. It simply means that the format changes you requested cannot be implemented on the current output data. These changes are implemented on the next data value that you enter into the output value field.

BASE - contains four choices (one is hidden), that allow you to enter output values into the output field in decimal, binary, octal, or hexadecimal format. The current selection is always bright. If the desired base is not bright, use the arrows at the bottom of the field to move the desired base to the center location. When the desired base appears in the center location, point to and select it to activate that base. The newly selected base will now be bright to indicate that it is the current selection.

Depending on the base, the legitimate values that you can enter into the output field are:

Decimal = values from -32768 to +65535
Octal = values from 0 to 177777
Hexadecimal = values from 0 to FFFF
Binary = values from 0000000000000000 to
                 1111111111111111

NOTE

When decimal base is selected, note that values from 0 to 65535 are interpreted as unsigned decimal numbers; values from −32768 to -1 are interpreted as 2's complement decimal numbers.

LENGTH · contains 16 choices (13 are hidden) that allow
you to specify the size of the output data segment
(number of output bits) that will be active. The length can
vary in size from 1 bit to 16 bits. Use the arrows at the
bottom of the field to move the desired choice into the
center location. When a selection appears in the center
location, it automatically becomes active. To prevent
errors, always specify the length of the field before you
specify the start bit. If errors do occur, a message in the
Status Window will identify the type of error that has
occurred. Unspecified output bits will be set low if the
logic sense is positive, or high if the logic sense is
negative.

START · contains up to 16 choices (13 are hidden) that let
you specify the starting location or LSB of your output
data segment. Because this field works in conjunction with
the length field, your start choices are limited by the
length of your field. If you specify a length of one bit, you
have a choice of 16 start bits. If you specify a length of 16
bits, you only have a choice of one start bit (bit 0). This
prevents the previously specified length of the data
segment from being truncated.

To illustrate how the size of your output data segment
affects the legitimate output values that you can enter into
the output field, consider the following example. Suppose
that you specified an output segment with length of 4 bits.
These four bits can be positioned anywhere within the 16
bit word with the start field. Depending on the base, the
legitimate values that you can enter into the output field
are:

    Decimal = integer values from −8 to +15
    Octal = values from 0 to 17
    Hexadecimal = values from 0 to F
    Binary = values from 0000 to 1111

LOGIC SENSE · formats the outgoing data as either
positive true logic, or negative true logic (output bits
inverted). When you point to and select this field, it
toggles between POSITIVE and NEGATIVE. The field

always indicates which logic sense is currently active. The output control signals are unaffected by this field.

For example, assume that you are going to enter a decimal value of 100 in the output value field. If the length of the data segment is 16 with a start bit of 0, a positive logic sense would set output bits 2, 5, and 6 high and all other bits low. A negative logic sense would set output bits 2, 5, and 6 low and all other bits high.

OUTPUT VALUE · consists of an entry field that lets you enter a value that determines the settings of the output bits. As previously discussed, the settings of the base, length, start, and logic sense fields will determine the values that you can enter into the output value field. To enter the output value, simply point to and select the output value field. Use the keyboard keys to position the cursor, type in the new value, and press ⌊Return⌋ . Note that the output bits are not set to the new value unless the output field indicates ENABLED.

DRIVER · lets you specify either an open collector output, or a TTL logic level output. When you point to and select this field, it toggles between TTL and OPEN COLLECT. The field always indicates which output type is currently active. Specifications for both output types are in Table 1-1.

| CAUTION |

To prevent damage to the instrument, disable the output whenever you change the driver.

An open collector output is used when output signals above 5 V are required, or when the output is used to power a device such as a LED or relay coil. An external power supply is required to drive the open collector outputs. Depending on your application, an external pull-up resistor or protection diode may also be required.

OUTPUT · this field toggles between output ENABLED and output DISABLED. The output field must indicate ENABLED for any data to be sent to an external device. Whenever the output is ENABLED, the data that is currently stored on the internal data registers of the Digital I/O is placed on the output bits. New data that is entered when the output is disabled will still be stored; however, it will not appear at the output until the output is enabled. When this occurs, the $\overline{ODAV}$ control signal pulses low for approximately $10\mu s$ to indicate that the output data has been sent. Output DISABLED deactivates all 16 output bits (into a tri-state condition). The **ENABLE OUTPUTS** and **DISABLE OUTPUTS** softkeys affect your instrument in the same way as output ENABLED/DISABLED.

Synchronous Operation

Operating your Digital Output synchronously requires that when output data is sent to an external device, the Digital Output signals the device that data is available. After the external device accepts the data, it returns a signal to the Digital Output to indicate it has successfully received the information. Two control signals — Output Data Available and Output Data Accepted — are available at the output connector, both of which must be connected to your external equipment (refer to Chapter 4). Once the connections are made, you are ready to operate your Digital Output synchronously.

The front panel graphics already discussed (i.e. output value, base, length, start, logic sense, driver and output fields) operate exactly the same way in synchronous mode as they do in asynchronous mode. The only additional fields that are used when operating your instrument synchronously are the handshake and reset fields (see Figure 3-5).

HANDSHAKE · must be ENABLED to operate your instrument synchronously. The Handshake field toggles between ENABLED and DISABLED. Point to and select DISABLED to enable the control signals that allow handshaking with your external equipment.
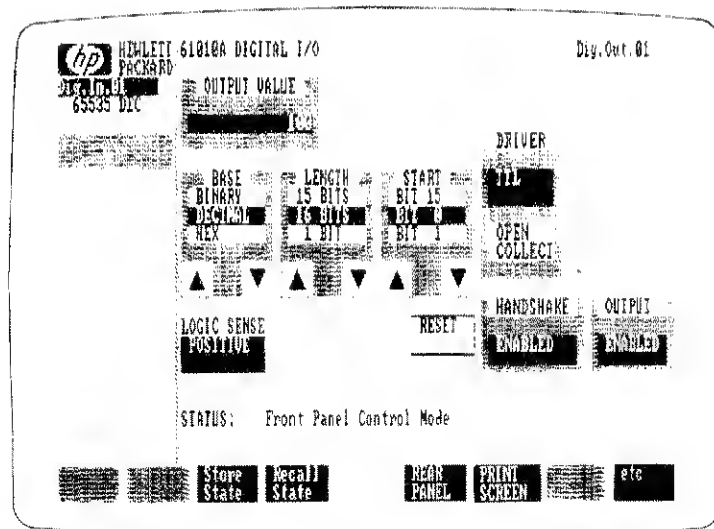
Figure 3-5. Handshake Enabled

RESET - when the handshake is enabled, the reset field appears on the screen only after a handshake sequence has begun. To start the handshake sequence you must enable the output, enter a value in the output field, and press ⌈Return⌋ . The reset field will now appear on the screen to indicate that the Digital Output is waiting for the external device to accept the data that has been sent. When the external device responds with an Output Data Accepted ($\overline{ODAC}$) signal, the reset field no longer appears on the screen. Point to and select the reset field when it appears on the screen to clear the handshake if an $\overline{ODAC}$ signal is not received. You must also clear the handshake before you enter any new data in the output value field.

Figure 3-6 shows the timing relationship between the Output Data Available ($\overline{ODAV}$) signal from the Digital Output and the Output Data Accepted ($\overline{ODAC}$) signal that must be provided by the external device.
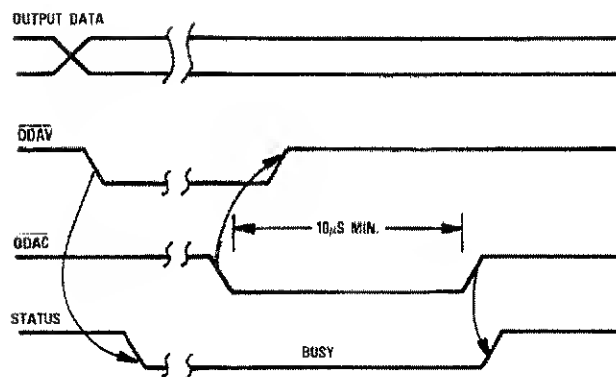
Figure 3-6. Output Handshake Timing Diagram

The high-to-low transition of $\overline{ODAV}$ alerts the external device that data is available. It also signals the status register that the Digital Output is busy, indicating that no further data should be sent. After the external device pulls $\overline{ODAC}$ low, indicating that the data is being accepted, $\overline{ODAV}$ goes high again. When the external device allows $\overline{ODAC}$ to go high again, indicating that all data has been accepted, the status register is informed that the Digital Output is ready for another transaction. The Digital Output cannot send more data to the external device until after $\overline{ODAC}$ is allowed to go high again by the external device.

4

Front Panel Connections

Introduction

⚠

Before you make any connections on the front of your instrument, you should already have connected your instrument to the interface card, connected the Power Pack and power cord, and set the address switch as explained in Chapter 2 of your System Owner's Manual. It is also recommended that you first try out your instrument as described in Chapter 3 of this manual before you make your application connections.

WARNING

To avoid potentially hazardous electrical shock, establish a safety ground before making any front panel connections. Connect the output cable from the Power Pack to the Digital I/O, and then connect the line cord from the Power Pack to the ac line (refer to Chapter 2 of the System Owner's Manual).

Application Connections

All application (i.e. field wire) connections to the Digital I/O are made on connectors that plug into the front of your instrument. Three connectors are supplied with each instrument. Use the female connector for all input application connections. Use one of the male connectors for all output application connections. The extra male connector is used in the verification procedures discussed in Appendix B.

The two connectors (one male, one female) that are used for application connections come equipped with all of the necessary hardware required to assemble them except the
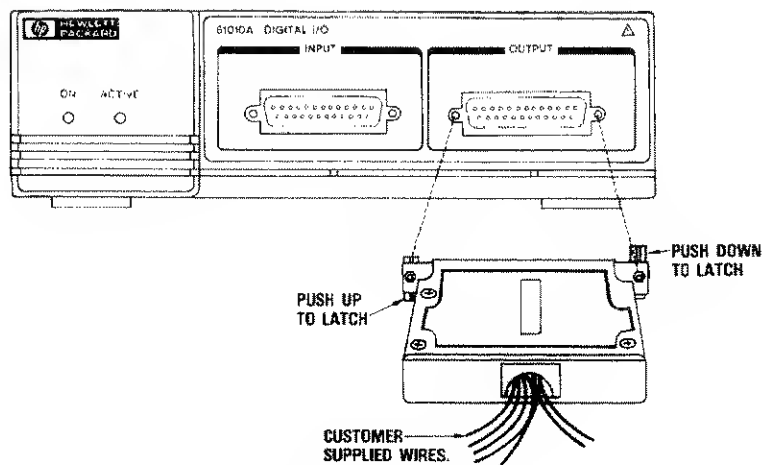
wires, solder, and tools, which you must provide. Figure
4-1 shows how an assembled connector is installed on the
front of the Digital I/O.

NOTE

*An optional Terminal Block (HP 14802A) is available that allows
you to connect wires to the Digital I/O using the screw
terminals it provides (see Appendix D).*

Figure 4-1. Application Connections

Assembling the Connector

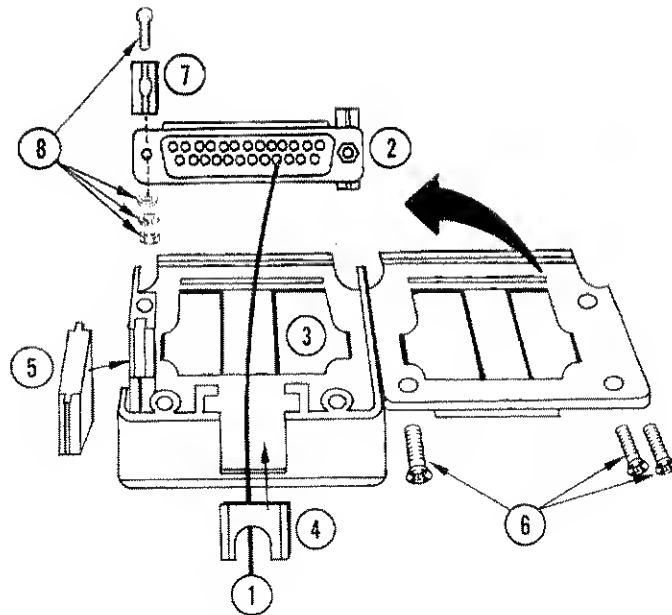Refer to Figure 4-2 and assemble the connectors that are
shipped with your instrument as follows:

CAUTION

*To prevent shorted connections when assembling your input and
output connectors, do not strip wires back more than 0.15 in
(4 mm). Use only AWG 20 (0.5 mm²) wire guage. Only use the
form and type of connector originally supplied with the
equipment.*

Figure 4-2. Exploded View of Connector Assembly

1. Solder the wires or cable (1) from your application to the solder pins on the connector (2). Figure 4-3 shows the location of the data bits and control signals on each connector.

2. Place the connector inside the cable clamp (3) and route the wires out of the opening at the back of the cable clamp.

3. Insert the proper strain relief (4) over the wires, and insert the solid plug (5) in the opening on the side of the cable clamp.

4. Close the cable clamp and tighten the three cover screws (6).

5. Attach the slide latches (7) to the connector using the hardware (8) that is supplied. The lip on the side of the slide latch must overlap the edge of the connector. Do not overtighten the screws.

6. Move the slide latches to the open position and insert your connectors in the mating sockets on the front of your instrument (see Figure 4-1).

7. Push the latches on each side of the connector to the locking position to securely fasten the connectors to the front of the instrument.

**FEMALE CONNECTOR
(CONNECTS TO INPUT)**

**MALE CONNECTOR
(CONNECTS TO OUTPUT)**
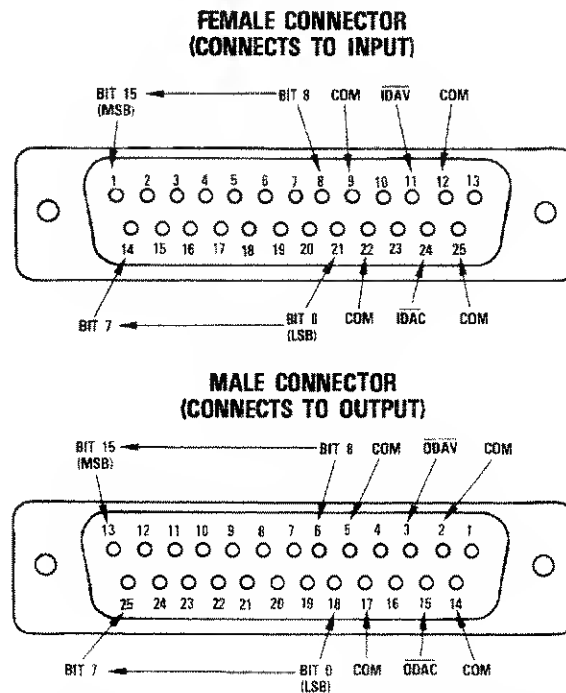
**Figure 4-3. Digital I/O Connector Pin Assignments
(viewed from the solder pin side of the connector)**

Single Stage Output Schematics

Figure 4-4 shows the output stage of one of the 16 output bits as well as the $\overline{\text{IDAC}}$ and $\overline{\text{ODAV}}$ control signals. With the TTL driver enabled, the output bits and control signals are compatible with TTL logic levels. When the TTL driver is disabled, the output bits and control signals are compatible with open collector logic levels. (Refer to the specifications in Table 1-1.)
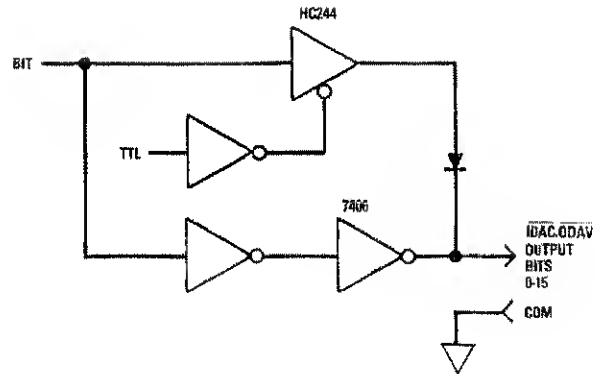
Figure 4-4. Output Data Bits and Output Control Signals

When open collector logic levels are required, you must provide an external pull-up resistor as well as an external supply voltage (see Figure 4-5). If your open collector output is connected to a device such as a relay coil, you must include a protection diode to prevent damage to the output stage due to voltage overshoot.

| CAUTION |

External bias voltages are limited to 12 volts.
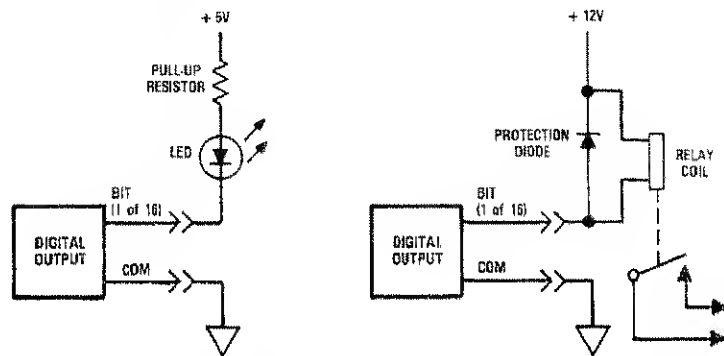
Figure 4-5. External Connections for Open Collector Outputs

Single Stage Input Schematics

| CAUTION |

Excessive input voltages will damage this instrument. Do not exceed ± 10 V max. per input bit.

Figure 4-6 shows the input stage for one of the 16 input bits and the input stage for the IDAV and ODAC control signals. The 16 input bits have a programmable threshold level. Chapter 3 describes how to set the threshold level. The control signals are compatible with TTL or CMOS logic levels. (Refer to the specifications in Table 1-1.) For best results when using your Digital Input, use pin 9 or 22 to make your data common connection. Control signal common connections can be made on pins 12 and 25.
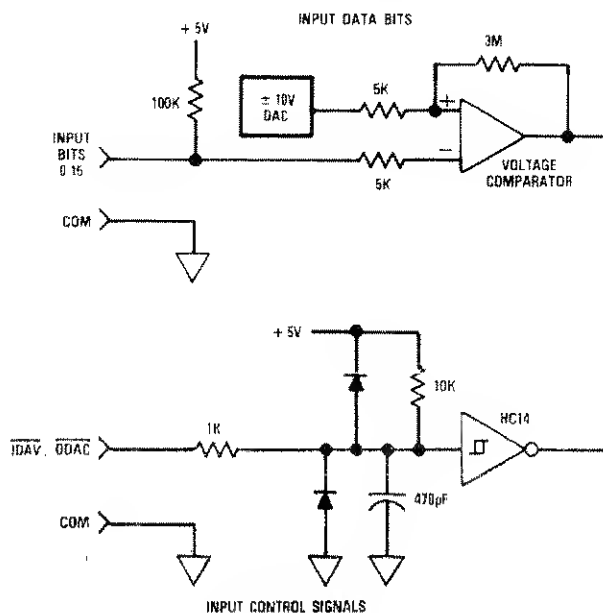
Figure 4-6. Input Data Bits and Input Control Signals

Connections for Synchronous Operation

Figure 4-7 illustrates the connections that are required to communicate or handshake with an external device during data transfers. Note that the Digital I/O requires a minimum pulse width of 10µs on the $\overline{\text{ODAC}}$ and $\overline{\text{IDAV}}$ signals that must be supplied by the external device. (Refer to Table 1-1 for further specifications.)
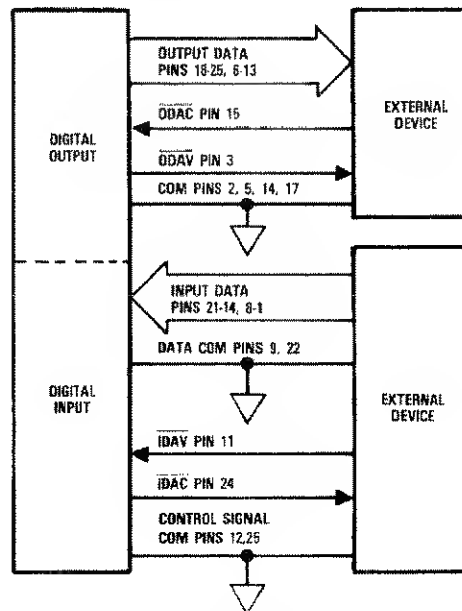
Figure 4-7. Handshake Connections for External Communication.

5

Programming With BASIC

General Information

You can write a BASIC program to perform read and write operations on your Digital I/O. Before you attempt this, you should familiarize yourself with controlling the instrument manually (refer to Chapters 2 and 3). Also, you must already know how to write programs in BASIC before you can write your own application program. Chapter 5 of the System Owner's Manual contains information about how to develop and run your program. Before writing your program, you must first use the Soft Front Panel to create a Program Shell as explained in Chapter 4 of the System Owner's Manual. You can also use the Soft Front Panel to assign labels to the Digital I/O and create one or more State files.

This chapter describes the statements that can be used in your program to control your Digital I/O. These statements fall into two categories: system and instrument. System statements affect other instruments in your system. Instrument statements only affect the specified (labeled) Digital Input or Digital Output.

NOTE

If the programming statements in this chapter fail to execute, you may have a program error. Refer to Chapter 5 in the System Owner's Manual which discusses error handling methods. Appendix C of this manual lists the error messages that apply to the Digital I/O.

How Statements Control the Digital I/O

Before using any of the statements in this chapter, take some time to review Figure 5-1. This block diagram illustrates the relationship between the programming statements that are discussed in this chapter and the functions of the Digital I/O. Because some of the data formatting is done with software and not in hardware, the statements that are implemented by software are surrounded by a dotted block to distinguish them from the statements that control the hardware.

From the diagram you can identify the two primary functions of the Digital I/O (i.e. Digital Input and Digital
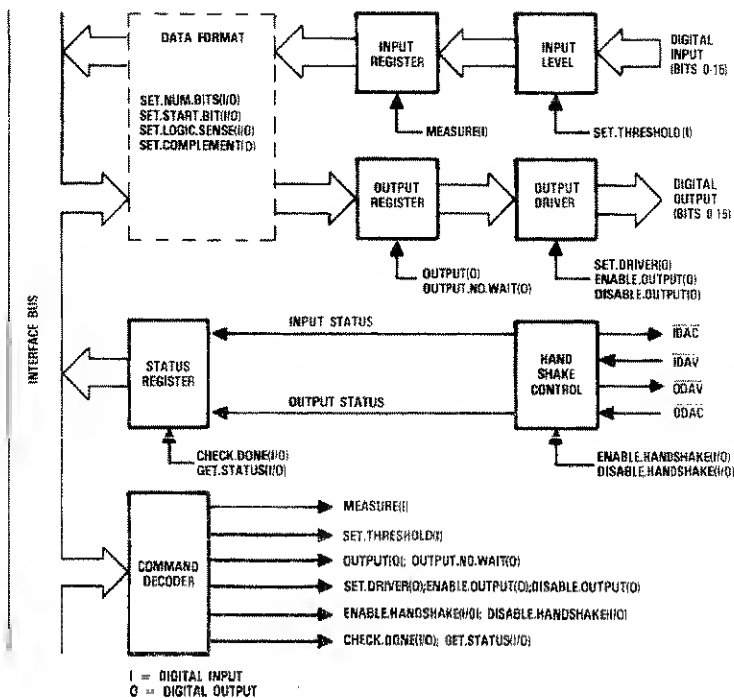
Figure 5-1. How Statements Control the Digital I/O

Output). Notice that some of the software formatting and hardware features are common to both the Digital Input and Digital Output. Because the Digital Input and Digital Output functions are independent of each other, you *must* specify whether you are programming either the Digital Input or the Digital Output by supplying the correct label after each statement. This is especially true when programming common features.

Some programming statements control functions that are specific to either the Digital Input or the Digital Output. For example, the MEASURE statement programs only the input register to read the data on the input bits. The OUTPUT statement loads only the output register with data. If the output driver is enabled by an ENABLE.OUTPUT statement, the output bits are set.

When the Digital I/O is operated synchronously, the ENABLE.HANDSHAKE statement activates the handshake control of either the Digital Input or the Digital Output. When the handshake control is enabled, the status of the control signals determines when data is either read by a MEASURE statement or sent by an OUTPUT statement. The CHECK.DONE statement checks the status of the control signals to determine if a handshake is in progress for either the Digital Input or the Digital Output. The INITIALIZE statement (not shown) sets the software functions and hardware registers for the Digital Input or the Digital Output according to a previously assigned State file.

System Programming Statements

All of the system programming statements are explained in Chapter 5 of the System Owner's Manual. Only three of them apply to the Digital I/O:

INITIALIZE.SYSTEM*(statefile)* - where *statefile* is a string variable that is equal to a State filename that you assigned when operating from the Soft Front Panel. This statement initializes both the Digital Input and Digital Output to the settings contained in the previously created State file. All other instruments in your system are also initialized. The INITIALIZE.SYSTEM statement should be used carefully because it affects all the settings of the Digital I/O. Whenever it is used in your program, it will override the results of any previously issued instrument statements. If this statement in not used in your program, the Digital I/O will be initialized with the factory default settings (see Table 1-1).

NOTE

The Rear Panel information contained in the State file must agree with your present hardware set-up. Rear Panel mode is described in Chapter 4 of the System Owner's Manual.

ENABLE.SYSTEM - enables the output bits on the Digital Output. This statement also enables any other output instruments in the system. It is recommended that you select a base, set the logic sense, number of bits, and the start bit before enabling the output. This ensures that the output data will be formatted correctly before it is applied or sent to the external device.

DISABLE.SYSTEM -disables the output bits on the Digital Output (into a tri-state condition). This statement also disables all other output instruments in your system.

Instrument Programming Statements

As explained in Chapters 2 and 3, the Digital I/O is identified by two labels, one for each of the primary functions of the instrument. In the following statements, the *label* parameter must specify either the Digital Input or the Digital Output. The label itself is either the factory default label (DIG.IN.01; DIG.OUT.01 for the first Digital I/O in your system) or the user-defined label that is assigned when the instrument is labeled in Rear Panel Mode. All of the examples in this section assume that the Digital Input and Digital Output are initially set with the factory default labels and values. There are three kinds of instrument statements that program the Digital I/O: common statements, input specific statements, and output specific statements. They are described as follows.

Statements Common to Input and Output

INITIALIZE*(label,statefile)* · same as INITIALIZE.SYSTEM except that it set either the Digital Input or the Digital Output to a previously assigned state. Although the State file contains information about the other instruments in your system, only the information that applies to either the Digital Input or the Digital Output will be retrieved.

Example:

```
1010 FILE$ = "YOURFILE"
1020 CALL INITIALIZE(DIG.IN.01,FILE$)
```

This example initializes only the Digital Input to the configuration and settings contained in the file YOURFILE. YOURFILE can be any State file that you created and named when operating from the Soft Front Panel.

ENABLE.HANDSHAKE*(label)* · enables the handshake mode of either the Digital Output or the Digital Input. This allows you to operate your Digital I/O synchronously with an external device. Refer to "Synchronous Operation" in Chapter 3 for a complete description.

Example:

    1010 CALL ENABLE.HANDSHAKE(DIG.IN.01)

This example enables the handshake mode only on DIG.IN.01.

DISABLE.HANDSHAKE*(label)* · disables the handshake mode of either the Digital Output or the Digital Input. This allows you to operate your Digital I/O asynchronously, where the computer alone controls data transmission.

Example:

    1010 CALL DISABLE.HANDSHAKE(DIG.OUT.01)

This example disables the handshake mode only on DIG.OUT.01.

CHECK.DONE*(label,flag)* · used in handshake mode to check the status of either the Digital Output or the Digital Input. The variable *flag* returns either a 0 or a 1. For the Digital Input, a 0 indicates that handshaking is done, and a 1 indicates that handshaking is in progress (busy). For the Digital Output, a 0 indicates that handshaking is in progress (busy) and a 1 indicates that handshaking is done. Figures 3-3 and 3-6 show when the status is busy during handshake mode.

Example:

    1010 CALL ENABLE.HANDSHAKE(DIG.IN.01)
    1020 CALL CHECK.DONE(DIG.IN.01,INSTATUS)
    1030 IF INSTATUS = 0 THEN GOTO 1020
    1040 CALL MEASURE(DIG.IN.01,VALUE)

In this example, line 1020 of the program checks the handshake status only of DIG.IN.01. Notice that INSTATUS is an example of a variable name that is meaningful within the context of this application. Line 1030 checks the value of the variable INSTATUS. If

INSTATUS = 0 it means that no signal has been received from the external device to indicate that input data is available. The program will continue looping until the handshake is busy, which indicates that new input data is available. At that time program execution continues at line 1040 which reads the input data with a MEASURE statement.

GET.STATUS*(label,value)* - same as CHECK.DONE except that it can also be used to determine if an input "overrun" has occurred (if the input data was changing faster than the computer could accept it). The following table explains the meaning of the values that are returned by the variable *value*.

Value	Digital Input	Digital Output
0	handshake done	handshake busy
1	handshake busy	handshake done
2	input overrun has occurred	
3	input overrun; handshake busy	

SET.LOGIC.SENSE*(label,polarity)* - applies to either the Digital Input or the Digital Output and lets you determine if the incoming or outgoing data is to be represented as positive true logic or negative true logic. The variable *polarity* must be specified as either POSITIVE or NEGATIVE. These are the only valid entries. The default polarity is positive.

Example:
1010 CALL SET.LOGIC.SENSE(DIG.OUT.01,NEGATIVE)

This example sets the outputs of DIG.OUT.01 to negative true logic.

SET.NUM.BITS*(label,bits)* - lets you specify the number of bits that will be used for data transactions on either the Digital Input or the Digital Output. The number of bits used are counted from the LSB (start bit) and proceed to the MSB. If the start bit is not specified, (see SET.START.BIT) the software will default to bit 0 as the start bit. The variable *bits* is a variable that must be set to one of the following reserved variable names: R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13, R14, R15, or R16. These variable names let you specify any number of bits from 1 to 16. For the Digital Output, unspecified output bits will be set low if the logic sense is positive, or high if the logic sense is negative.

Example:

```
1010 CALL SET.NUM.BITS(DIG.OUT.01,R8)
```

This example specifies eight output bits. Because a SET.START.BIT statement has not been programmed, the start bit of this data byte is bit 0. Therefore, output bits 0 through 7 are specified.

SET.START.BIT*(label,bit)* - lets you specify the start bit (LSB) of the data field that you specified by the SET.NUM.BITS statement on either the Digital Input or the Digital Output. To prevent programming errors, you should always program the SET.NUM.BITS statement before you program the SET.START.BIT statement. The SET.START.BIT statement can only be programmed if the number of bits of the specified data field is less than the default value of 16 bits. The variable *bit* is a variable that must be set to one on the following reserved variable names: R0, R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13, R14 or R15. These variable names let you specify any start bit from bit 0 to bit 15.

Example:

```
1010 CALL SET.NUM.BITS(DIG.OUT.01,R8)
1020 CALL SET.START.BIT(DIG.OUT.01,R8)
```

This example specifies bit 8 as the start bit of a data byte that consists of output bits 8 through 15. Line 1010 first specifies a total of eight output bits; line 1020 specifies the start bit.

NOTE

If you specify a start bit that truncates the length of the field previously specified by the SET.NUM.BITS statement, an error will occur (see Appendix C) and the operation will be ignored.

Statements Specific to Input

SET.THRESHOLD*(label,value)* · applies only to the Digital Input and sets a voltage threshold level for the incoming data on the Digital Input. Voltages above the threshold level are read as a logic "high". Voltages below the threshold level are read as a logic "low". The variable *value* is a single-precision variable that can be set to any voltage from −10.24 to +10.16 volts with a resolution of 0.08 volts.

Example:
```
1010 LEVEL = 2.56
1020 CALL SET.THRESHOLD(DIG.IN.01,LEVEL)
```

This example sets the voltage threshold level of DIG.IN.01 to 2.56 V.

SET.COMPLEMENT*(label,value)* · applies only to the Digital Input and determines whether decimal values will be represented in 2's-complement or unsigned form. The variable *value* must be specified as either TWOS or UNSIGNED. 2's-complement returns integers in the range of −32768 to 32767. Unsigned format returns numbers in the range of 0 to 65535.

Example:
```
1010 CALL SET.COMPLEMENT(DIG.IN.01,TWOS)
```

This example returns integers in the range of −32768 to 32767 when using a MEASURE statement.

MEASURE(*label,value*) · applies only to the Digital Input returns a reading of the data on the input bits into the variable *value*. The variable *value* will return decimal values in the range of −32768 to 65535. When 2's-complement is in effect, the range is from −32768 to 32767. When the format is unsigned, the range is from 0 to 65535. If the number of bits and the start bit specify an input field that is smaller than 16 bits, the unspecified bits are ignored by the MEASURE statement.

When the MEASURE statement is used with the handshake mode disabled, the input data register is updated with the current data on the input bits whenever the MEASURE statement is programmed. If the handshake is enabled, the MEASURE statement will wait until it has received an $\overline{\text{IDAV}}$, signal from the external device before it will read data into the input register.

Example:

```
1010 CALL SET.COMPLEMENT(DIG.IN.01,UNSIGNED)
1020 CALL MEASURE(DIG.IN.01,VOLTAGE)
1030 PRINT VOLTAGE
```

In this example, the data registers are programmed to read the data on the input bits. Line 1010 specifies an unsigned format, which returns decimal values in the range of 0 to 65535. Line 1020 causes the data on the input bits to be read into the computer. Notice that the variable name VOLTAGE is an example of using a variable name that can be meaningful within the context of a specific application. Line 1030 prints the value of the variable VOLTAGE.

Statements Specific to Output

SET.DRIVER(*label,value*) · applies only to the Digital Output and determine whether the output bits will be compatible with either TTL or open collector logic. The variable *value* must be specified as either TTL or OC. Chapter 4 gives examples of the external circuitry allowed when using open collector logic levels.

Example:

   1010 CALL SET.DRIVER(DIG.OUT.01,OC)

This example sets the output bits to be compatible with open collector logic levels.

ENABLE.OUTPUT*(label)* · same as ENABLE.SYSTEM except that it activates the output bits only the specified Digital Output.

Example:

   1010 CALL ENABLE.OUTPUT(DIG.OUT.01)

This example enables the output bits of DIG.OUT.01.

DISABLE.OUTPUT*(label)* · same as DISABLE.SYSTEM except that it deactivates (into a tri-state condition) the output bits of only the specified Digital Output.

Example:

   1010 CALL DISABLE.OUTPUT(DIG.OUT.01)

This example disables the output bits of DIG.OUT.01.

OUTPUT*(label,value)* · applies only to the Digital Output and programs the output register with the information contained in the variable *value*. The variable *value* must be a single-precision variable. It can only contain numbers in the following ranges:

   Decimal numbers from −32768 to 65535
   Hex numbers from 0 to FFFF (preceeded by &H)
   Octal numbers from 0 to 177777 (preceeded by &)

NOTE

These ranges only apply when all 16 bits are programmed, and the start bit is 0.

These numbers correspond to a binary range represented by the 16 output bits being programmed from all bits low to all bits high. The ENABLE.OUTPUT statement must be in effect for the output register to set the output bits according to the data programmed by the OUTPUT statement. If the handshake is disabled when the OUTPUT statement is programmed, the $\overline{\text{ODAV}}$ control signal strobes low for 10μs to indicate when the output bits are set. When the handshake is enabled, The OUTPUT statement will set the output bits, but will wait for a $\overline{\text{ODAC}}$ signal from the external device before allowing the program to continue to the next instruction (see Figure 3-6).

Example:

```
1010 CALL ENABLE.OUTPUT(DIG.OUT.01)
1020 VALUE = &HF
1030 CALL OUTPUT(DIG.OUT.01,VALUE)
```

In this example, line 1020 assigns a hexadecimal value of F to the variable *value*. (Hexadecimal F corresponds to a decimal value of 15 and a binary value of 1111.) In line 1030, the output register is programmed to set output bits 0 through 3 high. Note that the driver only activates the output bits if an ENABLE.OUTPUT statement has previously been programmed as shown in line 1010.

OUTPUT.NO.WAIT*(label,value)* · same as OUTPUT statement except that it does not wait for a $\overline{\text{ODAC}}$ signal from the external device. Program execution continues as soon as the output bits are set. This statement can be used in synchronous operation to send new data to the output bits without waiting for the $\overline{\text{ODAC}}$ signal to return (see Figure 3-6).

Example:

```
1010 CALL ENABLE.OUTPUT(DIG.OUT.01)
1020 CALL ENABLE.HANDSHAKE(DIG.OUT.01)
1030 VALUE = 15
1040 CALL OUTPUT.NO.WAIT(DIG.OUT.01,VALUE)
```

In this example, lines 1010 and 1020 enable the output and handshake signals. Line 1030 assigns the decimal value of 15 to the variable *value*. In line 1040, output bits 0 through 3 are set high. Program execution continues as soon as the output bits are set.

Sample Programs

Before you can try out the sample programs given here, you must first use the Soft Front Panel to generate a Program Shell. Then exit the Soft Front Panel and run PCIBAS. Load the Program Shell and type in the lines shown in the example after line 1000. Once you have finished typing in the sample program, save it and run it.

Normally you would also use the Soft Front Panel to save a State file for all of the instruments in your PC Instruments System. Then you could use an INITIALIZE statement in your program to set the Digital I/O to the settings that you previously saved in the State file. Because the INITIALIZE statement is not used in the sample programs here, your Digital Input and Digital Output will be set to the factory default settings (see Table 1-1). This means you must program each function of your Digital Input and Digital Output separately. When writing your program, remember to use the same labels in your instrument statements that you assigned to the Digital I/O when using the Soft Front Panel.

Digital Input Programs

The example programs that follow use the statements described in this section to input data on your Digital Input. They use the default label (DIG.IN.01), and assume that the default settings (see Table 1-1) are in effect. The first example is with the handshake disabled and the second example is with the handshake enabled.

Example 1

program shell

.

.

.

1010 LEVEL = 1.36
1020 CALL SET.THRESHOLD(DIG.IN.01, LEVEL)

These statements set the Digital Input's logic threshold level to 1.36 volts.

1030 CALL SET.COMPLEMENT(DIG.IN.01, TWOS)

This statement sets 2's-complement data format, meaning integers in the range of -32768 to 32767 are returned.

1040 CALL MEASURE(DIG.IN.01, ABCD)
1050 PRINT ABCD

These statements program the Digital Input to take a reading of the input bits. The result is stored in the variable ABCD, which is then printed out.

1060 STOP

Example 2

program shell

.

.

.

1010 LEVEL = 1.36
1020 CALL SET.THRESHOLD(DIG.IN.01, LEVEL)
1030 CALL SET.COMPLEMENT(DIG.IN.01, TWOS)
1040 CALL ENABLE.HANDSHAKE(DIG.IN.01)

These statements set the Digital Input's threshold, specify 2's-complement data format, and enable the handshake mode.

```
1050 FOR I=1 to 10
1060 CALL CHECK.DONE(DIG.IN.01,INSTATUS)
1070 IF INSTATUS=0 THEN GOTO 1060
```

These statements start a FOR/NEXT loop that first checks the handshake status to see if the $\overline{IDAV}$ signal has been sent by the external device.

```
1080 CALL MEASURE(DIG.IN.01,ABCD)
1090 PRINT ABCD
1100 NEXT I
```

When the signal is received, the input bits are read by the Digital Input and the loop repeats for the next reading of the input bits.

```
1110 STOP
```

Example 2 will execute faster without the CHECK.DONE routine used in lines 1060 and 1070. However, if the external device does not send the $\overline{IDAV}$ signal, the MEASURE statement will wait for it to arrive before reading any data from the input register. If the $\overline{IDAV}$ signal never arrives, it will be necessary for you to reset the computer to continue.

Digital Output Programs

The example programs that follow use the statements described in this section to output data on your Digital Output. They use the default label (DIG.OUT.01) and assume that the default settings (see Table 1-1) are in effect. The first example is with handshake disabled and the second example is with handshake enabled.

Example 1

program shell

```
.
.
.
```

```
1010 CALL SET.DRIVER(DIG.OUT.01,TTL)
```

This statement sets the Digital Output to be compatible with TTL logic levels.

```
1020 WXYZ = 15
1030 CALL OUTPUT(DIG.OUT.01,WXYZ)
```

These statements load data from the variable WXYZ into the data register. This instructs the data register to program output bits 0 through 3 high.

```
1040 CALL ENABLE.OUTPUT(DIG.OUT.01)
```

The output driver sets bits 0 through 3 high only after this statement is programmed.

```
1050 STOP
```

Example 2

program shell

.
.
.

```
1010 DIM A(10)
1020 CALL SET.DRIVER(DIG.OUT.01,TTL)
1030 CALL ENABLE.OUTPUT(DIG.OUT.01)
1040 CALL ENABLE.HANDSHAKE(DIG.OUT.01)
```

These statements dimension an array, set the Digital Output to be compatible with TTL logic levels, and enable the output driver and handshake mode.

```
1050 FOR I=1 to 10
1060 WXYZ=A(I)
1070 CALL OUTPUT.NO.WAIT(DIG.OUT.01,WXYZ)
```

These statements start a FOR/ NEXT loop that outputs data and does not wait for the $\overline{ODAC}$ signal to return from the external device.

```
1080 CALL CHECK.DONE(DIG.OUT.01,OUTSTATUS)
1090 IF OUTSTATUS=0 THEN GOTO 1080
1100 NEXT I
```

These statements check the handshake status to see when the $\overline{ODAC}$ signal is returned. When the signal is returned, the loop repeats for the next piece of output data.

```
1110 STOP
```

Example 2 will execute faster if an OUTPUT statement is used in line 1070 instead of the OUTPUT.NO.WAIT statement. Because the OUTPUT statement will wait for the $\overline{ODAC}$ signal to return before program execution continues, it is not necessary to include lines 1080 and 1090 when using an OUTPUT statement. However, if the external device does not return the $\overline{ODAC}$ signal when using an OUTPUT statement, it will be necessary for you to reset the computer to continue.

A

Programming Statement Summary

The following is a summary of the progamming statements that can be used to control the Digital I/O. You can use this summary as a reference guide for spelling and syntax of the available statements.

System Statements

DISABLE.SYSTEM
ENABLE.SYSTEM
INITIALIZE.SYSTEM*(statefile)*

Instrument Statements

Statements Common to Input and Output

CHECK.DONE*(label,flag)*
DISABLE.HANDSHAKE*(label)*
ENABLE.HANDSHAKE*(label)*
GET.STATUS*(label,value)*
INITIALIZE*(label,statefile)*
SET.LOGIC.SENSE*(label,polarity)* · where *polarity* must be one of the following reserved variables: POSITIVE or NEGATIVE
SET.NUM.BITS*(label,bits)* · where *bits* must be one of the following reserved variables: R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13, R14, R15, or R16
SET.START.BIT*(label,bit)* · where *bit* must be one of the following reserved variables: R0, R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13, R14, or R15

Statements Specific to Input

MEASURE*(label,value)*
SET.COMPLEMENT*(label,value)* - where *value* must be one of the following reserved variables: TWOS or UNSIGNED
SET.THRESHOLD*(label,value)*

Statements Specific to Output

DISABLE.OUTPUT(*label*)
ENABLE.OUTPUT(*label*)
OUTPUT(*label,value*)
OUTPUT.NO.WAIT(*label,value*)
SET.DRIVER(*label,value*) · where *value* must be one of the
following reserved variables: OC or TTL

B

Verification Procedures

Introduction

Verification procedures for your Digital I/O are included with your HP PC Instruments software. Instructions on how to load and run these procedures are given in Appendix B of your System Owner's Manual. When you run the verification program, step-by-step instructions appear on your computer's display to guide you through each procedure. This appendix specifies the required test equipment and briefly describes the tests performed on the Digital I/O.

Equipment Required

You must have an installed PC Instruments Interface Card and a PC Instruments Digital I/O. You will also need the third 25 pin, male D subminiature connector that was included with your Digital I/O connector package for verification. To perform the verification, you must install this male connector in the INPUT connector socket on the front of your Digital I/O. The other two connectors are used for your application connections. In addition to the connector, you must supply two clip leads to make the test connections from the input to the output connector socket.

What the Test Does

When you first run the verification program, it does a system level verification that partially tests all the instruments in your system. This part of the test checks that the interface and instruments respond to commands issued by the computer. This part of the test is described in Appendix B of your System Owner's Manual.

After your Digital I/O passes these preliminary tests, it will appear in a menu along with the other instruments in your system. This menu is the starting point for the instrument-specific verification tests for all of your instruments. To continue testing your Digital I/O, you must select it from this list and press the TEST softkey (f7). (When you select an instrument from this list, the ACTIVE indicator on the front of the instrument lights up.) Specifically, the Digital I/O verification checks three different functions of the instrument: input/output data transfer, the programmable input threshold, and the handshake control circuits.

Input/Output Data Transfer

This first test allows you to check each of the 16 output and input bits of the instrument one at a time. Each output bit is used to check the corresponding input bit (i.e. output bit 0 checks input bit 0, etc). The computer's screen displays the necessary test connections that you must make. When you have connected the specified input and output bits together with a clip lead, logic high and then logic low are programmed to the Digital Output and read by the Digital Input. If the proper values are read by the Digital Input, the test passes.

If there is a problem with either an input or an output bit, the test will fail.

Programmable Input Threshold

The input threshold, which is actually a reference voltage, is programmable for the Digital Input. To verify that this function is operating correctly, you must connect any one of the 16 input bits to any one of the 16 output bits. The program sets all the output bits to logic low, then reads all the input bits. The reference voltage is ramped from -1 V to $+1$ V to verify that the connected input bit reads a logic high when the reference voltage is greater than the input voltage (the logic low supplied by the output bit). Next, the 16 output bits are programmed to a logic high and the reference voltage is ramped from 4 V to 5.5 V to see at which voltage level the Digital Input properly reads a logic low.

If either of these two checks fail, then the test fails. However, the test can also fail if the selected input or output bit is not functioning correctly. If the test does fail, try a different input and output bit and repeat the test. This lets you determine if the problem is with the programmable input threshold or with a particular input or output bit.

Handshake Control Circuits

To perform the handshake control circuit test, you must connect the input handshake signals to the output handshake signals with the clip leads. The necessary connections are displayed on the computer's screen. The handshake signals are tested with the handshake enabled as well as with the handshake disabled by examining the contents of the Digital I/O's status register during input and output data transfers.

If there is a problem with any of the handshake signals, the test will fail.

In Case of Trouble

It is important that you complete all verification tests on both the system and instrument level. If you Digital I/O does not pass these tests, consult your PC Instruments Support Guide for information on the PC Instruments exchange program.

C   Error Messages

The following error messages apply to the Digital I/O.
When programming your instrument from BASIC, these
error messages are not only returned when you use the
error handling routine described in Chapter 5 of the
system Owner's Manual.

402 System error
404 Invalid output value
405 System error
406 Driver must be OC or TTL
407 Complement must be TWOS or UNSIGNED
408 Bit width invalid
409 Start bit invalid
410 System error
412 Threshold voltage out of range

D

HP 14802A Terminal Block Connections

The HP 14802A Terminal Block is designed to simplify wire connections to your HP PC Instruments Digital I/O. Each HP 14802A comes with two screw terminal fixtures and two cables to connect the fixtures to the front of your Digital I/O. One cable is terminated in a male connector, the other is terminated in a female connector. The male connector mates with the output connector on the Digital I/O front panel; the female connector mates with the input connector on the Digital I/O front panel. The two screw terminal fixtures are interchangeable and can connect to either cable. If desired, the screw terminal fixtures can be rail mounted. Figure D-1 shows how to connect this hardware to the front of your Digital I/O.

To make your wire connections to the screw terminal fixtures, first loosen all of the screws on top of the fixture. Strip your wires back approximately 0.2 in (5mm) and insert them in the square holes on the front of the terminal block. The screw terminals can accommodate wire sizes from AWG 14 to AWG 28 (Metric 1.5-0.5mm²). Check Tables D-1 and D-2 to make sure that you are inserting your wires into the correct opening. Table D-1 gives the terminal block assignments for the input connector. Table D-2 gives the terminal block assignments for the output connector.

CAUTION

To prevent shorted inputs, do not strip wires back more than 0.2 in. (5mm). Insert wires fully into screw terminals.

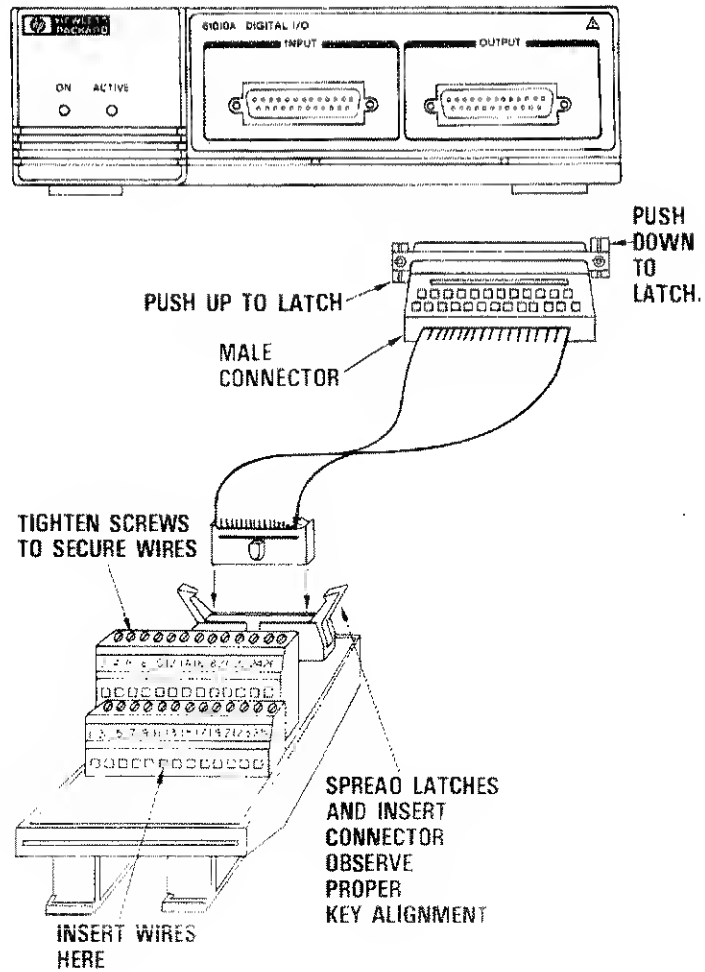---

Figure D-1. HP 14802A Connections

Table D-1. Input Terminal Block Assignments

Signal Name	Terminal Block#		Signal Name
bit 15	1	2	bit 7
bit 14	3	4	bit 6
bit 13	5	6	bit 5
bit 12	7	8	bit 4
bit 11	9	10	bit 3
bit 10	11	12	bit 2
bit 9	13	14	bit 1
bit 8	15	16	bit 0
common	17	18	common
N/C	19	20	N/C
$\overline{\text{IDAV}}$	21	22	$\overline{\text{IDAC}}$
common	23	24	common
N/C	25	26	N/C

Table D-2. Output Terminal Block Assignments

Signal Name	Terminal Block#		Signal Name
N/C	1	2	common
common	3	4	$\overline{\text{ODAC}}$
$\overline{\text{ODAV}}$	5	6	N/C
N/C	7	8	common
common	9	10	bit 0
bit 8	11	12	bit 1
bit 9	13	14	bit 2
bit 10	15	16	bit 3
bit 11	17	18	bit 4
bit 12	19	20	bit 5
bit 13	21	22	bit 6
bit 14	23	24	bit 7
bit 15	25	26	N/C